CSC 261/461
Database Systems

Eustrat Zhupa

October 24, 2018

UNIVERSITY of ROCHESTER

Functional Dependencies

Functional Dependency

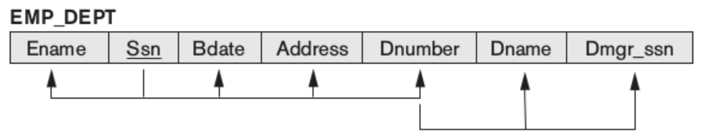
- $X \rightarrow Y$ holds if whenever two tuples have the same value for X, they must have the same value for Y
- For any two tuples t1 and t2 in any relation instance r(R): If $t1[X] = t2[X]$, then $t1[Y] = t2[Y]$
- $X \rightarrow Y$ in R specifies a constraint on all relation instances $r(R)$
- FDs are derived from the real-world constraints on the attributes

Functional Dependencies

Inferred FDs

If we denote by F the set of FDs that are specified on R.

- An FD $X \rightarrow Y$ is inferred from a set of dependencies F specified on R if $X \rightarrow Y$ holds in every legal relation state r of R.
- Given a set of FDs F, we can infer additional FDs that hold whenever the FDs in F hold.



Functional Dependencies

Armstrong's inference rules

- IR1. (Reflexive) If $Y \subseteq X$, then $X \rightarrow Y$
- IR2. (Augmentation) If $X \rightarrow Y$, then $XZ \rightarrow YZ$
- IR3. (Transitive) If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$
- IR1, IR2, IR3 form a `sound` and `complete` set of inference rules

Functional Dependencies

Other Inference Rules

- Decomposition: If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$
- Union: If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$
- Pseudotransitivity: If $X \rightarrow Y$ and $WY \rightarrow Z$, then $WX \rightarrow Z$
- The last three inference rules, as well as any other inference rules, can be deduced from IR1, IR2, and IR3 (completeness property)

Functional Dependencies

- Closure of a set F of FDs is the set F^+ of all FDs that can be inferred from F
- Closure of a set of attributes X with respect to F is the set X^+ of all attributes that are functionally determined by X
- X^+ can be calculated by repeatedly applying IR1, IR2, IR3 using the FDs in F.

Functional Dependencies

Algorithm for X^+

- **Input**: A set F of FDs on a relation schema R, and a set of attributes X, which is a subset of R.

$X^+ := X$
repeat
  $oldX^+ := X^+$
for each functional dependency $Y \to Z$ in F do
  if $Y \subseteq X^+$ then $X^+ := X^+ \cup Z$
until $(X^+ = oldX^+)$;

Functional Dependencies

Example

Consider the following relation schema about classes held at a university in a given academic year.

- CLASS (Classid, Course#, Instr_name, Credit_hrs, Text, Publisher, Classroom, Capacity).
- Let F, the set of functional dependencies for the above relation include:
  1. FD1: Classid $\rightarrow$ {Course#, Instr_name, Credit_hrs, Text, Publisher, Classroom, Capacity}
  2. FD2: Course# $\rightarrow$ Credit_hrs
  3. FD3: {Course#, Instr_name} $\rightarrow$ {Text, Classroom}
  4. FD4: Text $\rightarrow$ Publisher
  5. FD5: Classroom $\rightarrow$ Capacity

Functional Dependencies

Equivalent Sets

Two sets of FDs F and G are `equivalent` if:

- Every FD in F can be inferred from G, and
- Every FD in G can be inferred from F

Hence, F and G are equivalent if $F^+ = G^+$

Covers:

- F `covers` G if every FD in G can be inferred from F (if $G^+ \subseteq F^+$)
- F and G are equivalent if F covers G and G covers F

Functional Dependencies

Minimal Set of FDs

- A set of FDs is minimal if it satisfies the following conditions
  - Every dependency in F has a single attribute for its RHS.
  - We cannot remove any dependency from F and have a set of dependencies that is equivalent to F.
  - We cannot replace any FD $X \rightarrow A$ in F with a dependency $Y \rightarrow A$, where $Y \subset X$ and still have a set of FDs that is equivalent to F.

Functional Dependencies

Minimal Set of FDs

- A set of FDs is minimal if it satisfies the following conditions
  - A minimal set of FDs is a set of FDs in a standard or `canonical form` with no redundancies.
  - Condition 1 just represents every FD in a canonical form with a single attribute on the RHS.
  - Conditions 2 and 3 ensure there are no redundancies in the FDs either by having redundant attributes on the LHS of a dependency (Condition 2) or by having a dependency that can be inferred from the remaining FDs in F (Condition 3).
- A `minimal cover` of a set of FDs E is a minimal set of FDs that is equivalent to E.

Functional Dependencies

Finding a Minimal Cover F for a Set of Functional Dependencies E

- Input: A set of FDs E.

1. Set $F = E$.
2. Replace FD $X \rightarrow \{A1, A2, ..., An\}$ in F by the n FDs $X \rightarrow A1, X \rightarrow A2, \ldots, X \rightarrow An$.
3. For each functional dependency $X \rightarrow A$ in F
   for each attribute B that is an element of X
   if $\{\{F - \{X \rightarrow A\}\} \cup \{(X - \{B\}) \rightarrow A\}\}$ equiv. to F
   then replace $X \rightarrow A$ with $(X - \{B\}) \rightarrow A$ in F.
4. For each remaining FD $X \rightarrow A$ in F
   if $\{F - \{X \rightarrow A\}\}$ equiv. to F
   then remove $X \rightarrow A$ from F.

Functional Dependencies

Finding a Key K for R Given a set F of FDs

- Input: A relation R and a set of functional dependencies F on the attributes of R.
- Algorithm
    1. Set $K = R$.
    2. For each attribute A in K
        { compute $(K - A)^+$ with respect to F
        if $(K - A)^+$ contains all the attributes in R
        then set $K = K - \{A\}$}

Lossless Join Property

Lossless Join

- Let R be a relation schema
- Let F be a set of FDs on R.
- Let R_1 and R_2 form a decomposition of R.

The decomposition is a *lossless* decomposition if there is no loss of information by replacing R with two relation schemas R_1 and R_2.

```
select *
from  (select R1 from r)
     natural join
     (select R2 from r)
```

Nonadditive (Lossless) Join

Algorithm

Input: A universal relation R, a decomposition $D = R_1, R_2, \ldots, R_m$ of R, and a set F of FD.

1. Create an initial matrix S with one row i for each R_i, and one column j for each attribute A_j in R.
2. Set $S(i,j) = b_{ij}$ for all matrix entries. (distinct symbols)
3. For each row i representing relation schema Ri
   {for each column j representing attribute A_j
   {if (relation R_i includes attribute A_j)
   then set $S(i,j) = a_j$ }} (distinct symbols).

Nonadditive (Lossless) Join

Algorithm

4. Repeat until no changes to S

   {for each FD $X \to Y$ in F

   {for all rows in S that have the same symbols in the columns corresponding to attributes in X

   {make the symbols in each column that correspond to an attribute in Y be the same in all these rows as follows:
   If any of the rows has an a symbol for the column, set the other rows to that same a symbol in the column.
   If no a symbol exists for the attribute in any of the rows, choose one of the b symbols that appears in one of the rows for the attribute and set the other rows to that same b symbol in the column } }

   }

5. If a row is made up entirely of a symbols, then the decomposition has the nonadditive join property; otherwise, it does not.

Nonadditive Join Algorithm

Example

(a) $R = \{\text{Ssn, Ename, Pnumber, Pname, Plocation, Hours}\}$ $\qquad D = \{R_1, R_2\}$

$R_1 = \text{EMP\_LOCS} = \{\text{Ename, Plocation}\}$

$R_2 = \text{EMP\_PROJ1} = \{\text{Ssn, Pnumber, Hours, Pname, Plocation}\}$

$F = \{\text{Ssn} \rightarrow \text{Ename; Pnumber} \rightarrow \{\text{Pname, Plocation}\}; \{\text{Ssn, Pnumber}\} \rightarrow \text{Hours}\}$

	Ssn	Ename	Pnumber	Pname	Plocation	Hours
R_1	b_{11}	a_2	b_{13}	b_{14}	a_5	b_{16}
R_2	a_1	b_{22}	a_3	a_4	a_5	a_6

(No changes to matrix after applying functional dependencies)

Functional Dependencies

First Normal Form

(c) $R = \{\text{Ssn, Ename, Pnumber, Pname, Plocation, Hours}\}$ $\quad D = \{R_1, R_2, R_3\}$

$R_1 = \text{EMP} = \{\text{Ssn, Ename}\}$

$R_2 = \text{PROJ} = \{\text{Pnumber, Pname, Plocation}\}$

$R_3 = \text{WORKS\_ON} = \{\text{Ssn, Pnumber, Hours}\}$

$F = \{\text{Ssn} \rightarrow \text{Ename; Pnumber} \rightarrow \{\text{Pname, Plocation}\}; \{\text{Ssn, Pnumber}\} \rightarrow \text{Hours}\}$

	Ssn	Ename	Pnumber	Pname	Plocation	Hours
R_1	a_1	a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	a_3	a_4	a_5	b_{26}
R_3	a_1	b_{32}	a_3	b_{34}	b_{35}	a_6

(Original matrix S at start of algorithm)

	Ssn	Ename	Pnumber	Pname	Plocation	Hours
R_1	a_1	a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	a_3	a_4	a_5	b_{26}
R_3	a_1	$\cancel{b_{32}}$ a_2	a_3	$\cancel{b_{34}}$ a_4	$\cancel{b_{35}}$ a_5	a_6

Questions?